# AUTOMATICALLY BUILDING A SEARCHABLE DATABASE OF SOFTWARE FEATURES FOR SOFTWARE PROJECTS

## TECHNICAL FIELD

**[0001]** The present disclosure relates generally to software development. More specifically, but not by way of limitation, this disclosure relates to automatically building a searchable database of software features for software projects.

## BACKGROUND

**[0002]** Software projects are often stored in repositories accessible through websites, such as GitHub™. The repositories can contain files created throughout the lifecycle of the project. Examples of these files may include source code for the software project, code libraries, configuration files for installing or using the software project, readme files for describing aspects of the software project, and so on. These files may detail project requirements, functional requirements, design criteria, test data, and other information about the software project. Typically, the websites categorize software projects by keywords (e.g., topics or tags) in order to make identifying relevant software projects easier. The websites may also enable users to discuss the software projects (e.g., via a discussion forum), provide feedback to developers of the software projects, report bugs in the software projects, and download copies of the software projects.

## BRIEF DESCRIPTION OF THE DRAWINGS

**[0003]** FIG. **1** is a block diagram of an example of a system for automatically building a database of software features for software projects according to some aspects.

**[0004]** FIG. **2** is a block diagram of another example of a system for automatically building a database of software features for software projects according to some aspects.

**[0005]** FIG. **3** is a flow chart of an example of a process for automatically building a database of software features for software projects according to some aspects.

## DETAILED DESCRIPTION

**[0006]** Software developers may wish to discern if two software projects are sufficiently similar for various reasons, such as to use one software project as a replacement for another software project. But software projects often have hundreds or thousands of software features. A software feature is any functional characteristic of a software project, such as a framework relied on by the software project, a dependency (e.g., a code library or plugin) relied on by the software project, a programming language of the software project, other systems or software with which the software project is designed to interface, an executable format of the software project, types of algorithms implemented by the software project, etc. The large numbers of software features associated with any given software project makes it challenging or practically infeasible for a software developer to perform a meaningful comparison between software projects. These large numbers of software features also make it challenging for software developers to determine if software projects are fully compatible with their computing environments, since many of these software features can be defined internally and not explicitly described in relation to the software projects.

**[0007]** Some examples of the present disclosure overcome one or more of the abovementioned problems by automatically building a searchable database of software features for software projects. As one example, a system of the present disclosure can obtain descriptive information about a software project from one or more sources that may be internal or external to the software project. Examples of the descriptive information can include source code for the software project, configuration files or readme files provided with the software project, a description of the software project, or any combination of these. The descriptive information can be obtained from sources such as discussion threads on online forums, review websites, repositories, etc. The system can automatically analyze the descriptive information to determine software features of the software project. The system can then generate a feature vector for the software project based on the software features. A feature vector is a data structure containing elements, where each element is a numerical value indicating whether a particular software feature corresponding to the element is present or not present in a related software project. The data structure need not necessarily be a vector. After generating the feature vector, the system can store the feature vector in a database to enable the feature vector to be searched during a subsequent search process in response to a search query. The system can automatically repeat this process for dozens or hundreds of software projects to enable easy and fast searching and comparing of the software projects.

**[0008]** One simplified example of a feature vector may be {1, 1, 0, 0, 0, 0, 1, 0}, though typical feature vectors may have hundreds or thousands of elements. In this simplified example, the elements of the feature vector can correspond to the following exemplary software features, respectively: {python, machine-learning, web, django-framework, webassembly, sql, spark, gpu-support, java}. A binary value of 1 can indicate the presence of the software feature and a binary value of 0 can indicate the absence of the software feature, though other schemes are possible. In some instances, the numerical values of the elements may include decimal values, which can indicate a degree of likelihood that a software project has a particular software feature. The system can store a relationship between the feature vector and an identifier of its corresponding software project in the database. The feature vector can then be easily and quickly searched or compared to another feature vector to determine relationships between the corresponding software projects.

**[0009]** These illustrative examples are given to introduce the reader to the general subject matter discussed here and are not intended to limit the scope of the disclosed concepts. The following sections describe various additional features and examples with reference to the drawings in which like numerals indicate like elements but, like the illustrative examples, should not be used to limit the present disclosure.

**[0010]** FIG. **1** is a block diagram of an example of a system **100** for automatically building a database **116** of software features for software projects according to some aspects. The system **100** includes a computing device **102**, which may be a server, desktop computer, laptop computer, etc.

**[0011]** The computing device **102** can obtain descriptive information **104** about a software project **122**, such as a